DECOMPOSITION D'UN NOMBRE EN FACTEURS PREMIERS

Problème

Ce développement a déjà été publié dans la revue n°23. Nous le redonnons ici afin d'homogénéiser le dossier et de produire le listing du programme pour les anciennes calculatrices.

Principe

Un nombre ***A*** étant donné, on le divise par la suite des nombres premiers, en retenant ceux qui sont diviseurs, autant de fois que nécessaire, jusqu'à ce que le nombre premier soit supérieur à la racine carrée du dernier quotient entier.

La réélle difficulté d'une telle méthode est la génération de la suite des nombres premiers.
Pour ce faire, nous remarquerons qu'en dehors de ***2*** et de ***3***, tout nombre premier peut s'écrire sous l'une des formes:

6p + 1 ou ***6p + 5***.

Ce sont donc les nombres de cette forme que nous utiliserons comme diviseurs.
Ce qui ne signifie pas que tous ces nombres seront premiers (comme par exemple ***25 = 6*** x ***4 + 1***).
Cependant, nous détenons là un bon moyen de déterminer, sans trop de difficultés, tous les nombres premiers (plus quelques uns qui ne le seront pas). Nous effectuerons donc quelques divisions inutiles qui ne porteront aucun préjudice ni à la rigueur de la démarche ni à la rapidité d'obtention du résultat.

Génération des nombres ***6p + 1*** et ***6p + 5***.

Plutôt que de donner à ***p*** les valeurs 0, 1, 2, 3, etc… on remarquera, qu'à partir de 5, on obtient les suivants en:

- ajoutant **2** ***5 + 2 = 7***
- ajoutant **4** ***7 + 4 = 11***
- ajoutant **2** ***11 + 2 = 13***
- ajoutant **4** ***13 + 4 = 17***

etc…

et d'une manière générale, si on a le nombre 6p+1

- on ajoute **4**: 6p+1 + ***4*** = 6p+5 soit ***6p+5***
- on ajoute **2**: 6p+5 + ***2*** = 6(p+1)+1 soit ***6p+1***

etc…

Un problème se pose alors: comment obtenir successivement 2 puis 4, en ne faisant qu'une seule série d'opérations ?
N étant le diviseur, on appelle M la mémoire qui doit contenir 2 ou 4.
M vaut ***2***; en ajoutant ***2***, M vaut ***4***
M vaut ***4***; en ajoutant ***-2***, M vaut ***2.***
Il faut donc ajouter à M tantôt 2, tantôt -2.
Simulons cette situation:
N = 5 , M = 2 , J = -2:

- On augmente N de M (N+M→N) soit ***N=7***
- On prend l'opposé de J (-J→J) soit ***J=2***
- On augmente M de J (M+J→M) soit ***M=4***

- On augmente N de M (N+M→N) soit ***N=11***
- On prend l'opposé de J (-J→J) soit ***J=-2***
- On augmente M de J (M+J→M) soit ***M=2***

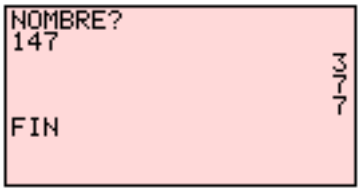
- On augmente N de M (N+M→N) soit ***N=13***
- On prend l'opposé de J (-J→J) soit ***J=2***
- On augmente M de J (M+J→M) soit ***M=4***

etc…

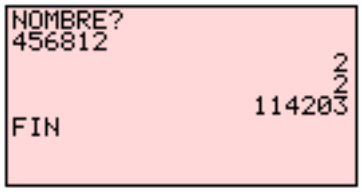
Utilisation

On lance le programme principal **FACT PRM**.

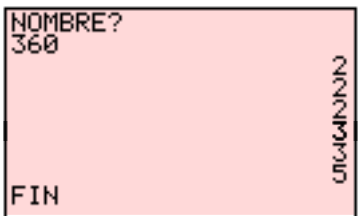
Exemple: ***147***
(on presse **EXE** après chaque affichage de diviseur)



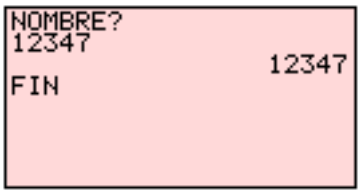
Exemple: ***456812***
(on presse **EXE** après chaque affichage de diviseur)



Exemple: ***360***
(on presse **EXE** après chaque affichage de diviseur)



Exemple: ***12347***
(on presse **EXE** après chaque affichage de diviseur)

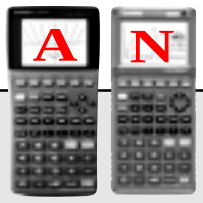


DECOMPOSITION D'UN NOMBRE EN FACTEURS PREMIERS

ALGORITHME:

• Demander A.
• N = 2
◊ *Procédure Divisions de A par N*
• N = 3
◊ *Procédure Divisions de A par N*
• N = 5 et M = 2 et J = -2
Faire:
  ◊ *Procédure Divisions de A par N*
  • Augmenter N de M
  • J devient -J
  • Augmenter M de J
Tant que N ≤ √A
• Si A≠1 alors écrire A et FIN
• Sinon FIN

◊ ***Procédure*** *Divisions de A par N:*
Tant que N divise A:
• Ecrire N
• Remplacer A par le quotient A/N
Fin de Tant que
FIN de procédure



N° du programme
Prog 5

```
'FACT 1
"NOMBRE"?→A
2→N
Prog 6
3→N
Prog 6
5→N:2→M:−2→J
Lbl 0
Prog 6
N+M→N
−J→J
M+J→M
N≤√A⇨Goto 0
A≠1⇨A
"FIN"
```

Nom du programme
FACT PRM

```
ClrText
"NOMBRE"?→A
2→N
Prog "DIVISE"
3→N
Prog "DIVISE"
5→N:2→M:-2→J
Do
Prog "DIVISE"
N+M→N
-J→J
M+J→M
LpWhile N≤√A
If A≠1
Then A
IfEnd
"FIN"
```

Nom du programme
Prog 6

```
'DIVISE
Lbl 2
A÷N→X
Frac X≠0⇨Goto 9
N
X→A
Goto 2
Lbl 9
```

Nom du programme
DIVISE

```
While Frac (A÷N)=0
N
A÷N→A
WhileEnd
```

DECOMPOSITION D'UN NOMBRE EN FACTEURS PREMIERS (suite)

Problème

Le programme de la fiche précédente présente de nombreux avantages:
- Pas très long à enregistrer.
- Exécutable sur les fx-6910G et GRAPH 20.
- Très performant dans la rapidité d'exécution des calculs.

Mais, son seul vrai défaut, est de donner la liste détaillée des différents facteurs premiers, par exemple 3; 3; 3; 3 au lieu de 3^4.

Nous allons donc arranger cela, dans cette fiche.
Cependant, comme nous allons exploiter les possibilités de mise en page qu'offre l'instruction `Locate`, ces programmes ne seront pas valides sur les fx-6910G et GRAPH 20.

Voir programmes au dos.

Utilisation

On lance le programme principal **FACT 1P**.

```
---DECOMPOSITION----
EN FACTEURS PREMIERS

NOMBRE?
610304
```

- *On affiche le nombre:* ***610304***
- *On presse* ***EXE***.

```
A=610304                FIN
2       ^12 :
149     ^1  :
            :
            :
            :
```

Les points simulent le partage de l'écran en deux colonnes

- *On lit:* ***610304 = 2^{12} x 149***
- *FIN indique que la décomposition est terminée.*

Autres exemples:

```
---DECOMPOSITION----
EN FACTEURS PREMIERS

NOMBRE?
126711200
```

```
A=126711200             FIN
2       ^5  :
5       ^2  :
7       ^1  :
11      ^3  :
17      ^1  :
            :
```

```
---DECOMPOSITION----
EN FACTEURS PREMIERS

NOMBRE?
407723916600
```

```
A=4.077239166E+11       FIN
2       ^3  :17     ^2
3       ^5  :29     ^1
5       ^2  :
7       ^1  :
11      ^1  :
13      ^1  :
```

```
---DECOMPOSITION----
EN FACTEURS PREMIERS

NOMBRE?
270364
```

```
A=270364                FIN
2       ^2  :
257     ^1  :
263     ^1  :
            :
            :
            :
```

```
---DECOMPOSITION----
EN FACTEURS PREMIERS

NOMBRE?
123457
```

```
A=123457                FIN
----------------------
     EST PREMIER
----------------------
            :
            :
            :
```

```
---DECOMPOSITION----
EN FACTEURS PREMIERS

NOMBRE?
1524323579
```

```
A=1524323579
12347   ^1  :
            :
```

Tant que le ***carré noir*** *est affiché, la calculatrice calcule.*
Et, … 4 minutes plus tard …, elle affiche le second facteur.

```
A=1524323579            FIN
12347   ^1  :
123457  ^1  :
            :
```

DECOMPOSITION D'UN NOMBRE EN FACTEURS PREMIERS (suite)

Indications

• On nomme le premier programme: **FACT 1P**.
• On nomme les sous-programmes (procédures):
- DIVISE1
- ECRAN
- PREMIER
- AFFICHE

Nom du programme

FACT 1P

```
ClrText↵
"---DECOMPOSITION----"↵
"EN FACTEURS PREMIERS"↵
" "↵
"NOMBRE"?→A↵
Prog "ECRAN"↵
2→N:0→r↵
Prog "DIVISE1"↵
3→N↵
Prog "DIVISE1"↵
5→N:2→M:-2→J↵
Do↵
Prog "DIVISE1"↵
N+M→N↵
-J→J↵
M+J→M↵
LpWhile N≤√A↵
If A≠1↵
Then If r=1↵
Then A→X:1→P↵
Prog "AFFICHE"↵
Else Prog "PREMIER"↵
IfEnd↵
IfEnd↵
Locate 19,1,"FIN"
```

Nom du programme

ECRAN

```
ClrText↵
2→L:0→C↵
Locate 1,1,"A="↵
Locate 3,1,A↵
For 2→Z To 7↵
Locate 11,Z,"."↵
Next
```

Ce sous-programme prépare l'écran pour un affichage des facteurs premiers sur deux colonnes.

Nom du programme

PREMIER

```
Locate 1,2,"--------------------"↵
Locate 1,3,"     EST PREMIER    "↵
Locate 1,4,"--------------------"
```

Ce sous-programme affiche que le nombre est premier (quand c'est la cas).

Nom du programme

DIVISE1

```
0→P↵
While Frac (A÷N)=0↵
N→X:P+1→P:1→W:1→r↵
A÷N→A↵
WhileEnd↵
If W=1↵
Then Prog "AFFICHE"↵
IfEnd
```

AFFICHE

```
If L>7↵
Then 2→L:11→C↵
IfEnd↵
Locate C+1,L,X↵
Locate C+8,L,"^"↵
Locate C+9,L,P↵
1+L→L:0→W
```

*Ce sous-programme affiche les uns sous les autres les facteurs premiers sous la forme **X^P**. Quand les (7-1) lignes d'affichage de la colonne de gauche sont prises, on passe à la colonne de droite (L=2 ; C=11).*